

**BLAKELY SOKOLOFF TAYLOR & ZAFMAN LLP**

TELEPHONE: (310) 207-3800

INTELLECTUAL PROPERTY LAW  
1279 OAKMEAD PARKWAY  
SUNNYVALE, CA 94085-4040FACSIMILE: (310) 820-5988  
(310) 820-5720**FACSIMILE COVER SHEET**

Deliver to: Examiner Taha, Shaq Art Group: 2122  
Company Name: USPTO  
Facsimile No.: 571-270-2921 Date: April 22, 2009  
From: Dimitri Kirmis, Reg. No.  
Our Docket No.: 6570P011 Number of pages      including this sheet.  
Application: 10/750,280 Filing Date: 12/30/2003

**Subject****Remarks**

Included in this fax is an amended set of claims which reflect the changes I would like to be made to the claims presented in the Examiner's Amendments of 04/13/09. Most of the changes can be easily seen in the amended set of claims. However, I have added commas which may not be readily visible. Specifically, commas have been added to the following portions of the claims:

Claim 12, Line 1;  
Claim 15, Line 5;  
Claim 21, Line 4;  
Claim 22, Line 5;  
Claim 25, Line 7; and  
Claim 31, Line 7.

Thank you for your help regarding this matter.  
Best Regards, Dimitri Kirmis

**Confidentiality Note:** The documents accompanying this facsimile transmission contain information from the law firm of Blakely, Sokoloff, Taylor & Zafman which is confidential or privileged. The information is intended to be for the use of the individual or entity named on this transmission sheet. If you are not the intended recipient, be aware that any disclosure, copying, distribution or use of the contents of this faxed information is prohibited. If you have received this facsimile in error, please notify us by telephone immediately so that we can arrange for the retrieval of the original documents at no cost to you.

**If you do not receive all the pages, or if there is any difficulty in receiving, please call: (310) 207-3800 and ask for Jessica M. Huaster.**

Attorney's Docket No. 6570P011

**EXAMINER'S PROPOSED CLAIM AMENDMENTS**

1. (Currently Amended) A method comprising:

establishing communication between a plurality of non-Java-based server nodes and a plurality of Java-based server nodes via an intermediate server, wherein Java-based server nodes are server nodes that implement Java 2 Platform Enterprise Edition (J2EE) applications;

generating a packet, on a first non-Java-based server node, to be transmitted from the first non-Java-based server node to a first Java-based server node;

specifying in a header of the packet an address of the first Java-based server node and information that indicates that the packet is generated by the first non-Java-based server node;

forwarding the packet to the intermediate server from the first non-Java-based server node;

forwarding the packet to the first Java-based server node from the intermediate server based on the address provided in the header of the packet, wherein the packet header is formatted to be compatible with the format of the first Java-based server node such that it can be decoded by the first Java-based server node, wherein the intermediate server providing interoperability of communications between the Java based server nodes and the non-Java based server nodes;

maintaining a list of services, the list of services includes processes and tasks performed by the Java-based server nodes; and

maintaining a list of services, the list of services includes processes and tasks performed by the non-Java-based server nodes, wherein the maintained lists of services facilitate communications between the first Java-based server nodes and ~~first the non-Java-based~~ server nodes.

2. (Currently Amended) The method of claim 1, further comprising:

generating a second packet to be transmitted from a second Java-based server node[[s]] to a second non-Java-based server node[[s]];

specifying in a header of the second packet an address of the second non-Java-based server node and information that indicates that the packet is generated by the second Java-based server node[[s]];

forwarding the second packet to the intermediate server from the second Java-based server node[[s]]; and

forwarding the second packet to the second non-Java-based server node from the intermediate server based on the address provided in the header of the second packet.

3. (Previously Presented) The method of claim 1 further comprising:  
sending notification of a status of each of the listed services to the non-Java-based server nodes such that the maintained lists of services can be updated.
4. (Currently Amended) The method of claim 1+3, further comprising:  
sending notification of a status of each of the listed services to the Java-based server nodes such that the maintained lists of services can be updated.
5. (Currently Amended) The method of claim 1+4, wherein the maintaining a list of services is accomplished by the intermediate server and the sending notification of a status of each of the listed services is accomplished by the intermediate server.
6. (Cancelled)
7. (Currently Amended) A system comprising:  
a plurality of non-Java-based server nodes, each of the non-Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from a first non-Java-based server node[[s]], wherein Java-based server nodes are server nodes that implement Java 2 Platform Enterprise Edition (J2EE) applications;  
a plurality of Java-based server nodes, each of the Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from a first Java-based server node; and  
a message server coupled between the Java-based server nodes and the non-Java-based server nodes to establish communication between the first Java-based server node and the first

non-Java-based server node, by formatting each packet header such that it is compatible with a format of a destination server; and

an enqueue server coupled between the Java-based server nodes and the non-Java based server nodes to provide central locking services to lock access to resources in the system for use during communications between the first Java-based server node and the first non-Java based server node such that communications between the first Java-based server node and the first non-Java based server node are not interrupted.

8. (Currently Amended) The system of claim 7, further comprising a dispatcher to distribute client requests for services to the Java-based server nodes and the non-Java based server nodes ~~server nodes of the respective instance~~, such that the dispatcher distributes the load amongst the Java-based server nodes and the non-Java based server nodes during communications.

9. (Previously Presented) The system of claim 7, wherein the message server provides communications between the Java-based server nodes and the non-Java based server nodes by routing message packets between the non-Java-based server nodes and the Java-based server nodes.

10. (Previously Presented) The system of claim 7, wherein the message server is to assign a service identification associated with each type of services executed on the server nodes, wherein assigning the service identifications assists in providing communications between the Java-based server nodes and the non-Java based server nodes.

11. (Previously Presented) The system of claim 10, wherein the message server includes a service repository to maintain a list of the assigned service identification and corresponding service names, wherein maintaining the service identifications and the service names assists in providing communications between the Java-based server nodes and the non-Java based server nodes.

12. (Currently Amended) The system of claim 8, wherein the message server further comprises:

a first repository to maintain a list of services currently being executed on the non-Java-based server nodes; and

a second repository to maintain a list of services currently being executed on the Java-based server nodes, wherein the first and second repositories facilitate distribution of load by indicating currently executing services.

13. (Currently Amended) The system of claim 8, wherein the message server is to maintain a list of services performed by the Java-based server nodes and the non-Java based server node[[s]] and a status corresponding to each of the listed services, and to send notification of the status of the listed services to the dispatcher such that the dispatcher distributes the load in each instance following requests for services.

14. (Cancelled)

15. (Currently Amended) A message server comprising:

a first communication interface to establish communication with a plurality of non-Java-based server nodes;

a second communication interface to establish communication with a plurality of Java-based server nodes, wherein Java-based server nodes are server nodes that implement Java 2 Platform Enterprise Edition (J2EE) applications; and

a controller to transfer packets between the non-Java-based server nodes and the Java-based server nodes, the controller to ensure the packets are received by a destination server node by resending the packets if a confirmation of receipt has not been received from the destination server node.

16. (Previously Presented) The message server of claim 15, wherein the controller is to assign a service identification associated with each type of services executed on the Java based server nodes and non-Java-based server nodes, wherein assigning the service identifications assists in providing communications between the first and second communications interfaces.

17. (Previously Presented) The message server of claim 16, further comprising:

a service repository maintain a list of the assigned service identification and corresponding service names, wherein maintaining the service identifications and the service names assists in providing communications between the first and second communications interfaces.

18. (Previously Presented) The message server of claim 15, further comprising:  
a first repository to maintain a list of services currently being executed on the non-Java-based server nodes; and

a second repository to maintain a list of services currently being executed on the Java-based server nodes, wherein the first and second repositories facilitate distribution of load by indicating currently executing services.

19. (Previously Presented) The message server of claim 17, wherein the controller is to send notification of a status of each of the services listed in the first repository to the non-Java-based server nodes such that the maintained lists of services can be updated.

20. (Previously Presented) The message server of claim 17, wherein the controller is to send notification of a status of each of the services listed in the second repository to the Java-based server nodes such that the maintained lists of services can be updated.

21. (Currently Amended) A machine-readable medium that provides instructions, which when executed by a processor cause the processor to perform operations comprising:  
establishing communication with a plurality of non-Java-based server nodes;  
establishing communication with a plurality of Java-based server nodes, wherein Java-based server nodes are server nodes that implement Java 2 Platform Enterprise Edition (J2EE) applications;  
transferring packets between the non-Java-based server nodes and the Java-based server nodes; and  
ensuring the packets are received by a destination server node by resending the packets if a confirmation of receipt has not been received from a destination server node.

22. (Currently Amended) The machine-readable medium of claim 21, wherein the operations performed by the processor further comprise:

assigning a service identification associated with each type of services executed on the server nodes; and

maintaining a list of the assigned service identification and corresponding service names, wherein maintaining the service identifications and the service names assists in providing communications between the non-Java-based server nodes and the Java-based server nodes.

23. (Previously Presented) The machine-readable medium of claim 21, wherein the operations performed by the processor further comprise:

maintaining a list of services currently being executed on the non-Java-based server nodes in a first repository; and

maintaining a list of services currently being executed on the Java-based server nodes in a second repository, wherein each list facilitates distribution of load by indicating currently executing services.

24. (Previously Presented) The machine-readable medium of claim 23, wherein the operations performed by the processor further comprise:

sending notification of a status of each of the services listed in the first repository to the non-Java-based server nodes such that the maintained lists of services can be updated; and

sending notification of a status of each of the services listed in the second repository to the Java-based server nodes such that the maintained lists of services can be updated.

25. (Currently Amended) A system comprising:

means for generating a packet on a first non-Java based server node such that a header of the packet specifies an address of a first Java-based server node, wherein Java-based server nodes are server nodes that implement Java 2 Platform Enterprise Edition (J2EE) applications;

means for indicating that the packet is generated by a non-Java based server node;

means for forwarding the packet to intermediate communication means from the first non-Java-based server node[[s]], wherein the intermediate communication means provides~~server~~

providing interoperability of communications between the Java based server nodes and the non-Java based server nodes; and

means for forwarding the packet to the first Java-based server node from the intermediate communication means based on the destination address provided in the header of the packet;

means for maintaining a list of services, the list of services includes processes and tasks performed by the first Java-based server node, wherein maintaining the list of service identifications and the service names assists in providing communications between the first non-Java-based server node and the first Java-based server node; and

means for sending notification of a status of each of the listed services to the first Java-based server node, the notification indicates whether the service is running or stopped such that the maintained list of services can be updated.

26. (Previously Presented) The system of claim 25, further comprising:

means for generating a second packet on a second Java based server node such that a header of the second packet specifies an address of a second non-Java-based server node and that the second packet originated from a Java-based server node[[s]];

means for forwarding the second packet to the intermediate communication means from the second Java-based server node; and

means for forwarding the second packet to the second non-Java-based server node from the intermediate communication means based on the destination address provided in the header of the second packet.

27. (Previously Presented) The system of claim 26, wherein the intermediate communication means further comprises:

means for maintaining a list of services performed by the Java-based server nodes wherein maintaining the list assists in providing communications between the Java-based server nodes and the non-Java-based server nodes; and

means for sending notification of a status of each of the listed services to the Java-based server nodes such that the maintained list of services can be updated.



28. (Previously Presented) The system of claim 27, wherein the intermediate communication means further comprises:

means for maintaining a list of services performed by the non-Java-based server nodes wherein maintaining the list of service identifications and the service names assists in providing communications between the non-Java-based server nodes and the Java-based server nodes; and

means for sending notification of a status of each of the listed services to the non-Java-based server nodes such that the maintained list of services can be updated.

29. (Previously Presented) The system of claim 25, wherein the intermediate communication means further comprises:

means for establishing communication with a plurality of non-Java-based server nodes;

means for establishing communication with a plurality of Java-based server nodes; and

means for transferring packets between the non-Java-based server nodes and the Java-based server nodes.

30. (Previously Presented) The system of claim 25, wherein the intermediate communication means further comprises:

means for assigning a service identification associated with each type of services executed on the Java based server nodes and the non-Java based server nodes; and

means for maintaining a list of the assigned service identification and corresponding service names, wherein maintaining the service identifications and the service names assists in providing communications between the Java-based server nodes and the non-Java-based server nodes.

31. (Currently Amended) A system comprising:

a plurality of non-Java-based server nodes, each of the non-Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to specify that the packet originated from a first non-Java-based server node;

a plurality of Java-based server nodes, each of the Java-based server nodes executing software instructions to attach a header to a body of a packet, the header including information to

specify that the packet originated from a first Java-based server node, wherein Java-based server nodes are server nodes that implement Java 2 Platform Enterprise Edition (J2EE) applications;

a message server coupled between the Java-based server nodes and the non-Java-based server nodes to establish communication between the first Java-based server node and the first non-Java-based server node by formatting each packet header such that it is compatible with a format of a destination server, the message server comprising:

a first repository to maintain a list of processes and tasks performed by the plurality of non-Java based server nodes, the first repository includes a plurality of rows each associated with a service performed by a non-Java based server and a plurality of columns which identify attributes associated with each service, the attributes include a server identification to indicate a server performing the service, a service mask to identify the type of service, and a status to indicate the status of the service;

a second repository to maintain a list of processes and tasks performed by the plurality of Java based server nodes, the second repository includes a plurality of rows each associated with a service performed by a Java based server and a plurality of columns which identify attributes associated with each service, the attributes include a server identification to indicate a server performing the service, a service mask to identify the type of service, and a status to indicate the status of the service; and

a third repository to maintain a list of assigned service identifications and their corresponding service names, the third repository includes a plurality of rows each associated with a service and a plurality of columns which identify attributes associated with each service, the attributes include a service name to indicate the name of the service and an assigned service mask to indicate the type of service.